MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2016/2017

TCP1241 – COMPUTER PROGRAMMING II

(All sections / Groups)

8 OCT 2016
2.30 p.m – 4.30 p.m
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 9 pages, including the cover page, with 5 Questions only.

2. Attempt **ALL THREE** questions in **Section A.**

3. Attempt **ONE out of TWO** questions in **Section B**. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

SECTION A: Please attempt **ALL THREE questions.**

Question 1

a. Describe the term of abstraction. How is the abstraction applied in the C++ class? [2 marks]

b. Based on the class `Test` below, fill up the right column of the table according to the types of the members specified on the left column. [3 marks]

```
class Test {
      int x;
      static int y;
  public:
      Test();
      Test (int a);
      Test (const Test&);
      ~Test();
      int getX();
      static int getY();
      void setX (int x);
};
```

Overloaded constructor(s):	
Copy constructor(s):	
Destructor(s):	
Update method(s):	
Query method(s):	

c. List THREE (3) differences between a constructor and a method. [3 marks]

d. Identify the problem of the multiple inheritance. Suggest a way to avoid the problem. [2 marks]

Continued .....

Question 2

a. Write the class declaration by using the Class Diagram below. [3 marks]

Exam
- code : string # category: string
+ Exam (code: string, category: string) + getCode () : string - funcExam (string: code) : void

b. Write a class called `Receipt` for an item sold at the store with the following attributes and methods. [3 marks]

i. Attribute `item` of type string, attribute `quantity` of type integer and attribute `price` of type double.

ii. Default constructor that initializes attribute `item` to "name", attribute `quantity` to `0` and attribute `price` to `0.1`.

iii. Overloaded constructor that initializes all attributes.

c. Refer to the code below, write the method definition for function `setAddress()` and `getAddress()` and place them outside the class declaration. [2 marks]

```
class Location {
      string address;

      public:
        void setAddress(string address);
        string getAddress();
};
```

Continued .....

d. What is the output of the following program? [2 marks]

```
class Log{
      static int count;
   public:
      Log(){ count+=2; }
      static int getCount() { return count; }
      ~Log(){ count-=2;}
};

int Log::count = 0;

int main(){
   cout << "Initial Count: " << Log::getCount()<<endl;
   Log L1;
   cout << "Current Count: " << Log::getCount()<<endl;
   {
       Log L2;
       cout << "Current Count: " << L2.getCount()<<endl;
   }
   cout << "Final Count: " << Log::getCount() << endl;
}
```

Continued .....

Question 3

a. Rewrite and transform the following class definition into an abstract class. [2 marks]

```
class Memo{
     public:
          void message() { cout<<"Meeting"; }
};
```

b. Explain the reason of transforming a class into an abstract class. [1 mark]

c. Given the program below,

```
class First{
    int a;
    public:
        First(){   a = 1;  }
};

class Second: protected First{
    public:
        Second(){   a = 2;  }
        void display(){ cout<<"a = "<<a; }
};

int main(){
    Second s;
    s.display();
}
```

i. What is the output after executing the program above? Justify your answer. [2 marks]

ii. If you are only allowed to modify class `First`, how would you change to fix the error? [1 mark]

Continued .....

d. Complete the class definition for class Exam, so that the program would be able to run and produce the output 5. [2 marks]

```
class Exam{
     int num;
```

```
};

void showID (Exam a){      cout << a.num << endl;      }

int main(){
    Exam a(5);
    showID(a);
}
```

e. Rewrite and complete the classes below to produce the output as shown. [2 marks]

```
class X {
     . . .
};

class Y: public X {
     . . .
};

int main() {  //Do Not modify the main()
  X *p = new X;
  p->display();
  delete p;
  p = new Y;
  p->display();
}
```

Output:

```
aa
cc
```

Continued .....

SECTION B: Please attempt **ONE** out of **TWO** questions only.

Question 4

a. Construct the class definitions for the 3 classes that have the attributes and methods as follow: [5 marks]

Class Employee:
Attribute: *private string name*
Method: *public overloaded constructor*
Class Permanent:
Inherits from class Employee
Attribute: *private double allowance*
Method: *public overloaded constructor*
Class Admin:
Inherits from class Permanent
Attribute: *none*
Method: *public overloaded constructor*

b. Change the following program using Exception Handling. The program decides the fee to be paid by the user based on the number of hours. [3 marks]

```
int main(){
  double hour;
  cout << "Number of hours: ";
  cin >> hour;
  if (hour < 1)
    cout << "Minimum number of hours is 1. Try again.\n";
  else
    cout << "Fee is RM"<<hour*2<<".\n";
}
```

c. Apply a function template named `print()` to the following program. The sample output is commented in the program as below. [2 marks]

```
#include <iostream>
using namespace std;

int main(){
    char c = 'X';
    double d = 3.67;
    print( c );    //output = X
    print( d );    //output = 3.67
}
```

Continued .....

Question 5

a. When overloading an operator, the number and the type of parameters depend on whether the operator is unary or binary and is being declared as a friend function or as a class method. Complete the method declarations for the following overloaded operators.

[2 marks]

Example: Overloading operator+ as friend function for adding two objects of class `Rational`.

```
friend Rational operator+ (Rational&, Rational& );
```

i. Overloading operator* as class method for multiplying two objects of class `ABC`.

ii. Overloading operator>> as friend function for inputting a `Rational` object.

b. Based on the given main function, complete the class definitions for class `State`, class `Area`, and class `Unknown` with proper declaration for attributes and constructors. Class `Area` inherits from class `State`; whereas class `Unknown` inherits from class `Area`.

[5 marks]

```
int main(){
    State s(10);
    Area a(10,100);
    Unknown u(10,100,1000);
}
```

Continued .....

c. Modify the main function by using vector instead of array. [3 marks]

```
class Visitor {
  string vname;
 public:
  void setName (string name) { vname = name; }
};

int main() {
  int size;
  Visitor* v;
  string name;
  cout << "Number of visitors: ";
  cin >> size;
  v = new Visitor[size];
  for (int i = 0; i < size; i++) {
    cout << "Name of visitor : ";
    cin >> name;
    v[i].setName (name);
  }
  delete [] v;
}
```

End of Paper